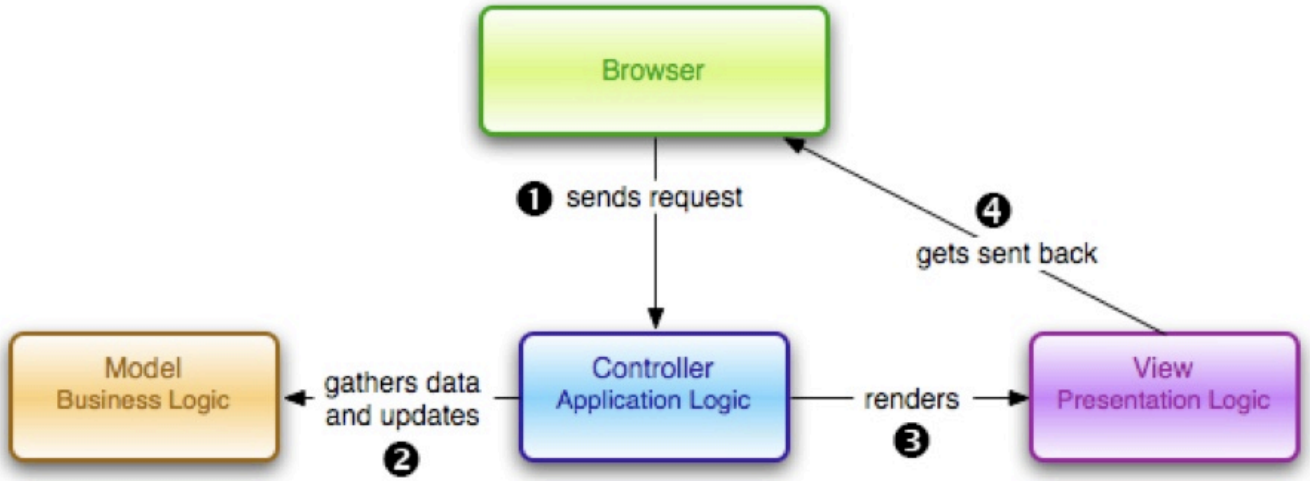
Models, Views, and Controllers
Lenx - Chapter 5

Charles Severance

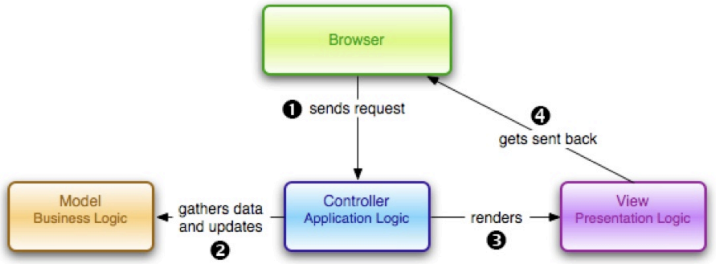
Textbook: Build Your own Ruby on Rails Application by Patrick Lenz (ISBN:978-0-975-8419-5-2)

MVC - Review


Browser
❶ sends request
❹
gets sent back
Model
Business Logic
gathers data
and updates
❷
Controller
Application Logic
renders
❸
View
Presentation Logic

MVC Sequence

- User presses button, browser sends data to application
- Controller receives the data, and makes updates to and/or retrieves from the model as necessary
- User output data is passed to the View - view applies final look and feel and the response goes back to the Brower.

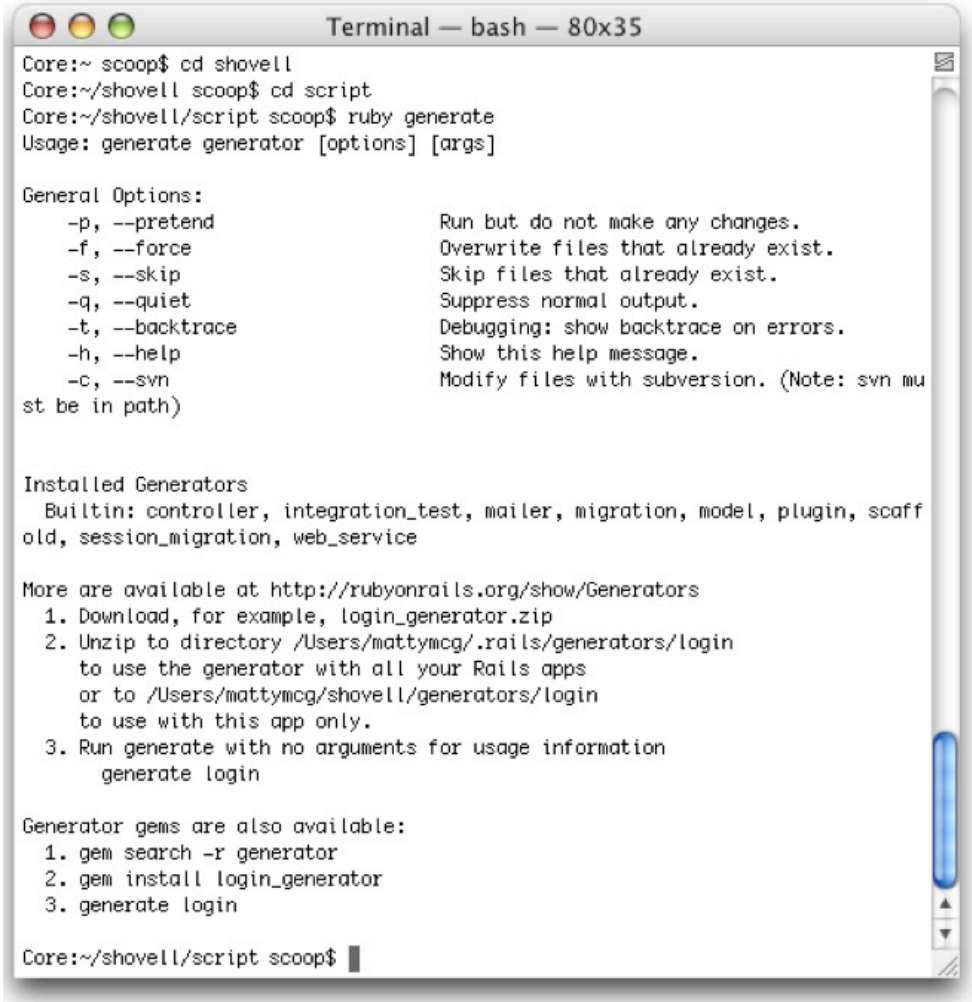


Time to Write Some Code

- Ruby uses a pattern called “generators”
- Generators write empty program elements for us to extend, improve and customize
- This is scaffolding for new developers and helps improve programmer productivity by automating common tasks
- You *could* write all this code by hand - there is nothing magical in the generators

The Generator

- ruby script/generate
- shows help output



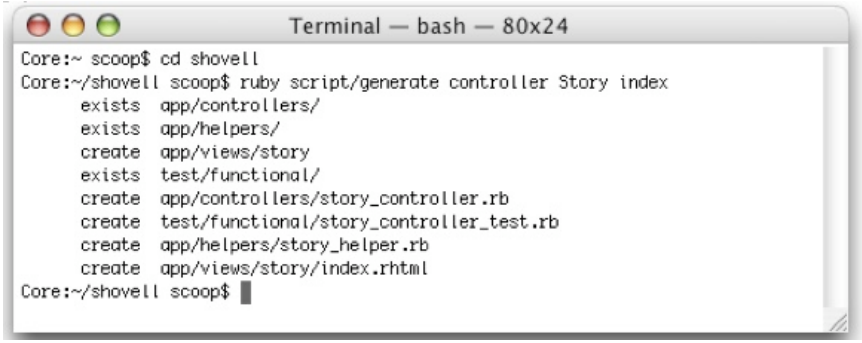
Making a Controller

Name the controller Story

- ruby script/generate controller Story index

Make a controller

Add action named index (default)



Controller Files

- story_controller.rb - The controller itself with a skeleton controller inside
- story_controller_test.rb - A blank placeholder for tests you might build
- story_helper.rb - A blank place for helper code
- index.rhtml - Your view file with a skeleton view inside

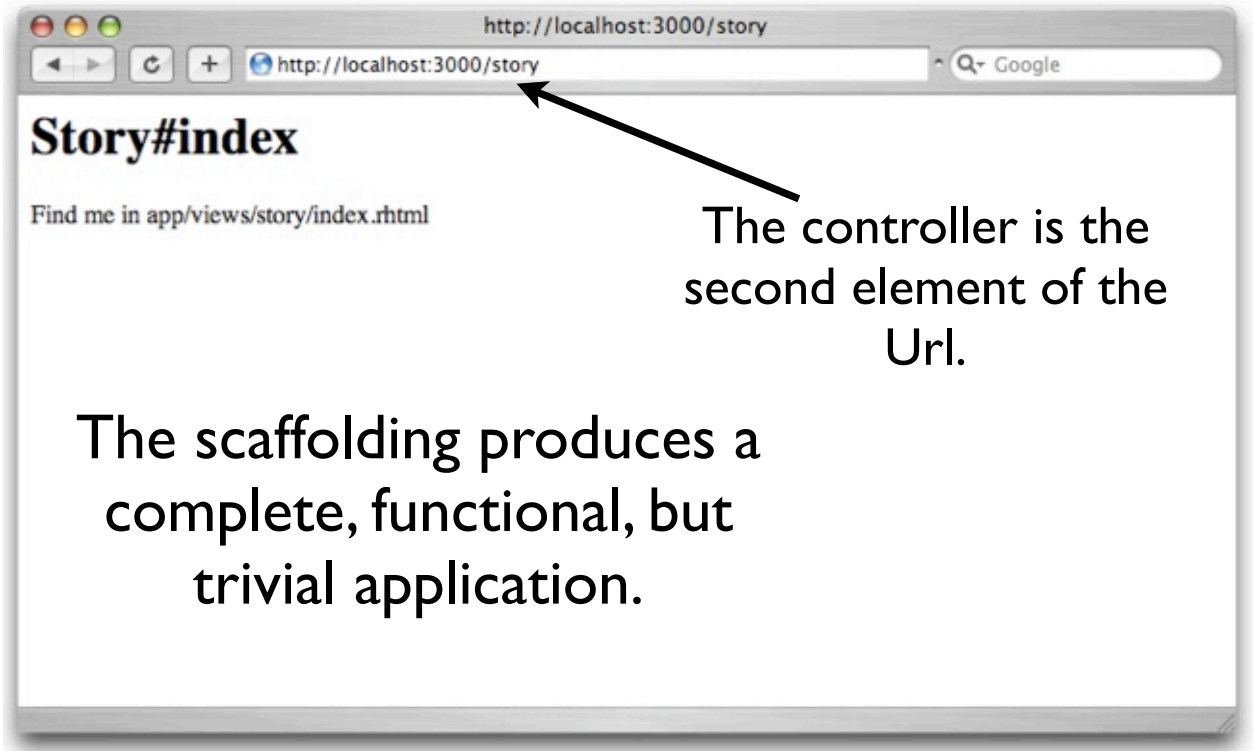
Scaffolding Controller and View

app/controllers/story_controller.rb

```
class StoryController < ApplicationController
  def index
  end
end
```

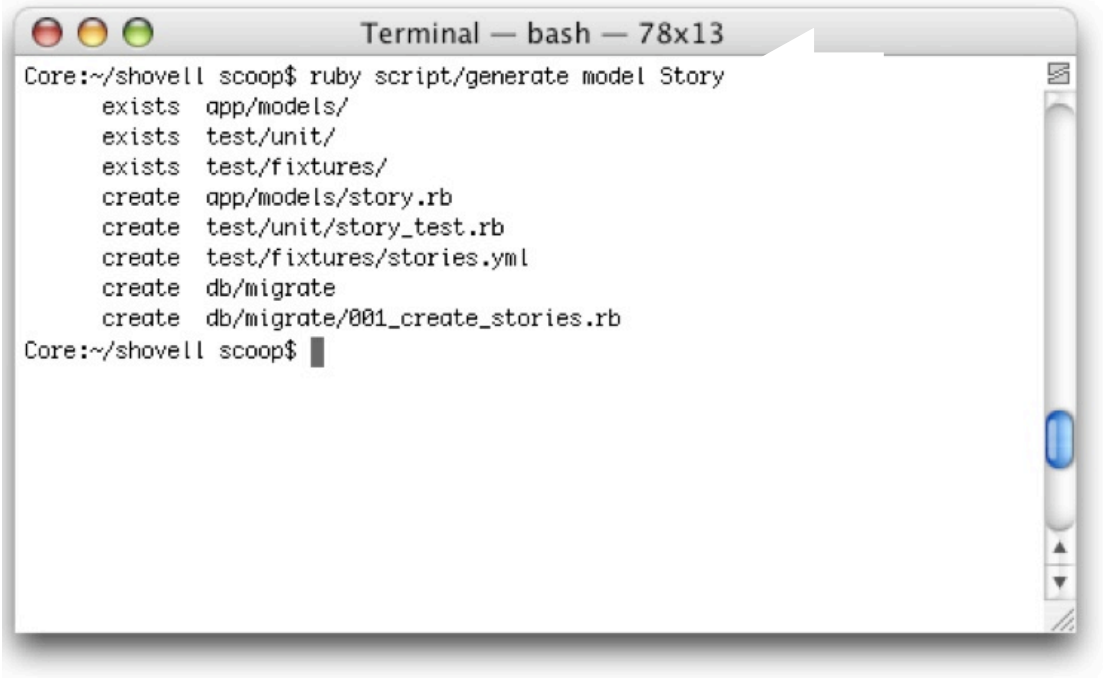
app/views/story/index.rhtml

```
<h1>Story#index</h1>
<p>Find me in app/views/story/index.rhtml</p>
```



The scaffolding produces a complete, functional, but trivial application.

ruby script/generate model Story



Model Files

- story.rb - The definition of the class that is the model that we use in Rails - it contains "Story" which we access in x = Story.create()
- story_test.rb - A place to put our code to test our model
- stories.yml - A way to pre-populate data in database tables - primarily used for testing
- 001_create_stories.rb - A "migration" - this defines a table structure in the database - later migrations can evolve the table structure as new fields are added or other structure changes are made

Yet Another Markup Language

- A way to make objects and set properties on those objects.
- Without using XML :)
- Primary use is ot pre-populate your database tables when building unit tests

stories.yml

```
# set initial values for the stories table
first:
  id: 1
  name: My shiny weblog
  link: http://poocs.net/
another:
  id: 2
  name: SitePoint Forums
  link: http://www.sitepoint.com/forums/
```

* Also a play on words - yml comes after xml

Migrations

- The concept of a table structure that evolves over time
  - Version 0 - no table at all
  - Version 1 - The first version
  - Version 2 - Making some changes

http://www.youtube.com/watch?v=n1NVfDIU6yQ

001_create_stories.rb

- self_up method - what to do to go from version 0 to version 1
- self_down - what to do to go from version 1 to version 0
- When there is more than on migration, they get chained together in numeric order

```
class CreateStories < ActiveRecord::Migration
  def self.up
    create_table :stories do |t|
      t.column :name, :string
      t.column :link, :string
    end
  end
  def self.down
    drop_table :stories
  end
end
```

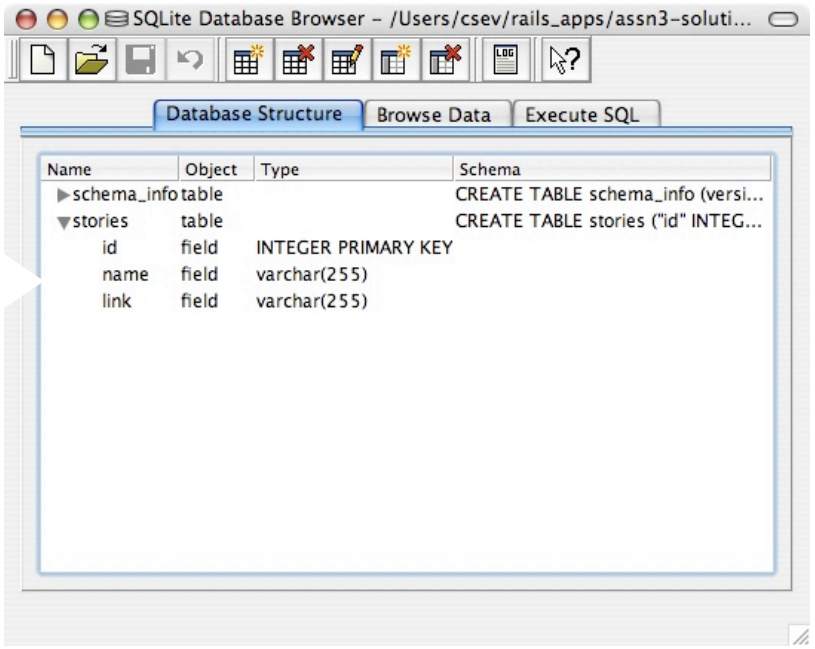
Causing Migrations to Happen

- rake db:migrate
  - Evolve the database from current version to highest version
- rake db:migrate VERSION=0
  - Evolve from the current version down to zero - generally no database at all.

Note - The text is wrong - you don't need a "-" before VERSION

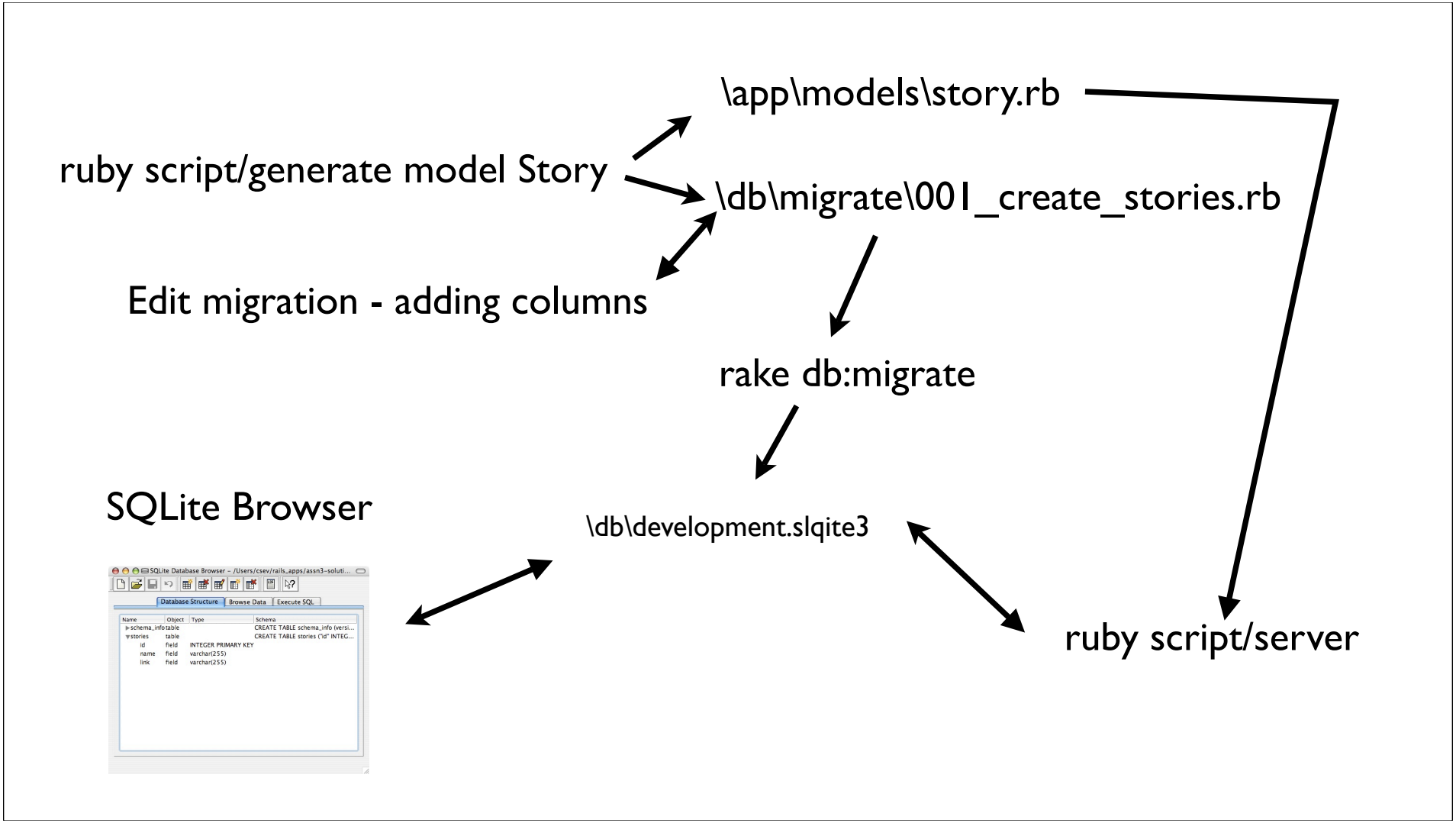
After rake db:migrate

```
class CreateStories < ActiveRecord::Migration
  def self.up
    create_table :stories do |t|
      t.column :name, :string
      t.column :link, :string
    end
  end
  def self.down
    drop_table :stories
  end
end
```



Step By Step - Database Setup

- Make your model - ruby script/generate model Story
- Edit your migration file - add columns - 001_create_stories.rb
- Run rake db:migrate
- Check your database in SQLite Browser
  - \db\development.sqlite3
- The model is now ready to use in Rails once you restart your server with ruby script/server


\app\models\story.rb
ruby script/generate model Story
\db\migrate\001_create_stories.rb
Edit migration - adding columns
rake db:migrate
SQLite Browser
\db\development.slqite3
ruby script/server

SQLite Browser Tips

- Make sure you are opening the right file - when you are switching between projects you may be looking at the wrong database
- To see new information in the browser - re-open the file
- If you make a change in the browser without saving the file will be locked and Rails will be unhappy
- You *can* make changes in the browser - just press save afterwards - for example - you can delete a record in the browser to test add over and over

The Rails Application Console

- Like Interactive Ruby (irb) but with the ActionRecord models loaded and ready to use - it is equivalent to being in a Controller method - but interactive

```
$ ruby script/console
Loading development environment.
>> i = 1
=> 1
>> i = i + 1
=> 2
>> s = Story.new
=> #<Story:0x3355388
@new_record=true,
@attributes={"name"=>"NULL",
"link"=>"NULL"}>
>>
```

Making Models by Hand

```
$ ruby script/console
Loading development environment.
>> s = Story.new
=> #<Story:0x285718c
@new_record=true,
@attributes={"name"=>nil,
"link"=>nil}>
>> s.name = 'My shiny weblog'
=> "My shiny weblog"
>> s.link = 'http://poocs.net/'
=> "http://poocs.net/"
>> s.save
=> true
```

```
>> s.id
=> 1
>> s.new_record?
=> false
>> Story.create(
:name => 'SitePoint Forums',
:link => 'http://www.sitepoint.com/forums/')
=> #<Story:0x279d474 @new_record=false,
@errors=#<ActiveRecord::Er
rors:0x279c72c @errors={}, @base=#<Story:
0x279d474 ...>>, @attribu
tes={"name"=>"SitePoint Forums", "id"=>2,
"link"=>"http://www.site
point.com/forums/"}>
```

Retrieving Models

```
>> Story.find(2)
=> #<Story:0x2796ca0 @attributes={"name"=>"SitePoint Forums",
   "id"=>"2", "link"=>"http://www.sitepoint.com/forums/"}>
>> Story.find(:all)
=> [#<Story:0x2788f9c @attributes={"name"=>"My shiny weblog",
   "id"=>"1", "link"=>"http://poocs.net/"}>, #<Story:0x2788e70
   @attributes={"name"=>"SitePoint Forums", "id"=>"2",
   "link"=>"http://www.sitepoint.com/forums/"}>]
>> Story.find(:all).last
=> #<Story:0x277f80c @attributes={"name"=>"SitePoint Forums",
   "id"=>"2", "link"=>"http://www.sitepoint.com/forums/"}>
```

Sorting and Retrieving by name

```
>> Story.find(:first, :order => 'id DESC')
=> #<Story:0x2779100 @attributes={"name"=>"SitePoint Forums",
   "id"=>"2", "link"=>"http://www.sitepoint.com/forums/"}>

>> Story.find_by_name('My shiny weblog')
=> #<Story:0x2773bd8 @attributes={"name"=>"My shiny weblog",
   "id"=>"1", "link"=>"http://poocs.net/"}>
```

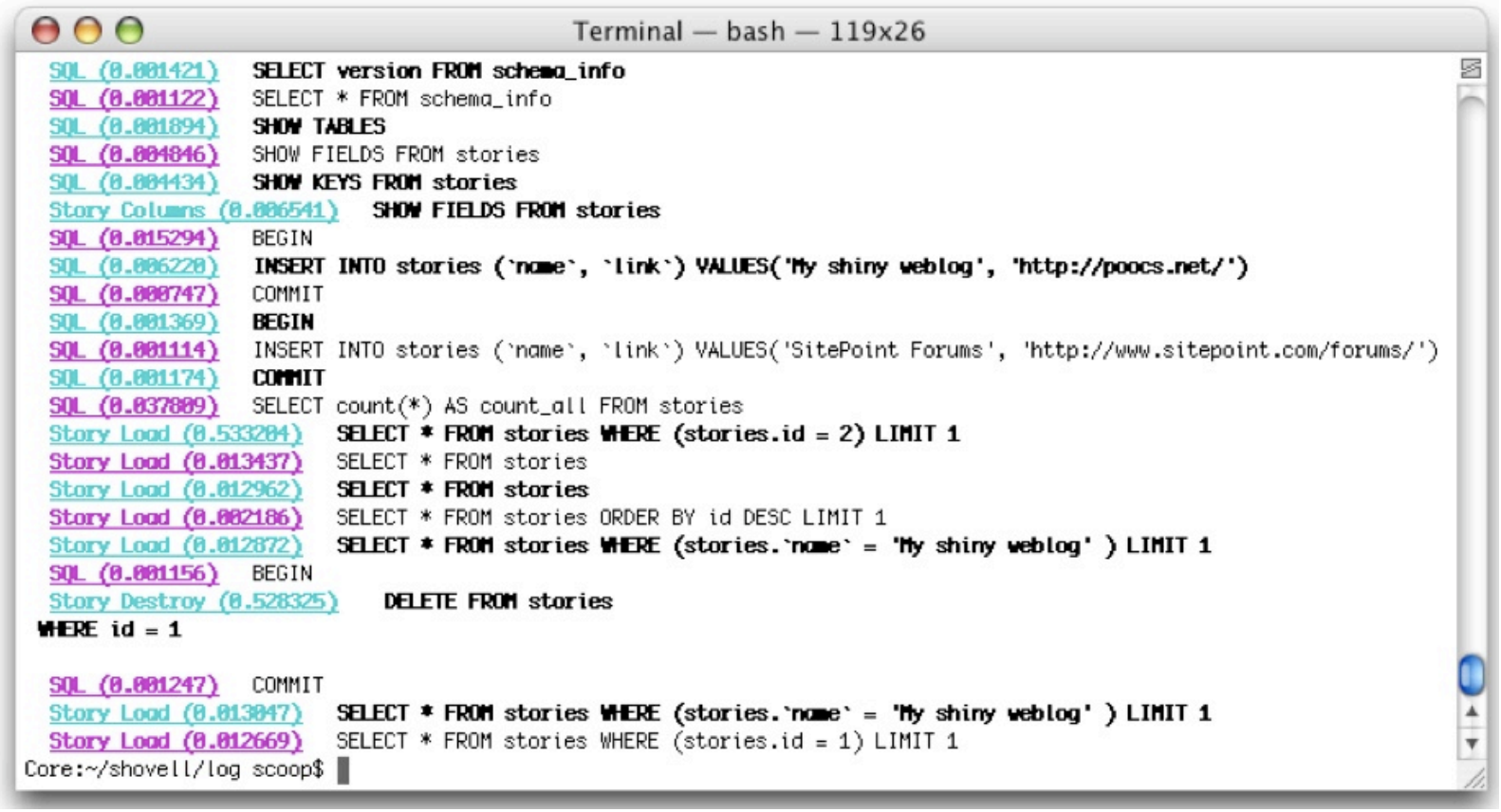
Updating the Database

```
>> s = Story.find_by_name('My shiny weblog')
=> #<Story:0x272965c ...>
>> s.name
=> "My shiny weblog"
>> s.name = 'A weblog about Ruby on Rails'
=> "A weblog about Ruby on Rails"
>> s.save
=> true

>> s.update_attribute :name, 'A weblog about Ruby on Rails'
=> true
```

Deleting Records

```
>> s = Story.find_by_name('My shiny weblog')
=> #<Story:0x272965c ...>
>> s.name
=> "My shiny weblog"
>> s.destroy
=> #<Story:0x272965c ...>
```


Terminal — bash — 119x26
SQL (0.001421) SELECT version FROM schema_info
SQL (0.001122) SELECT * FROM schema_info
SQL (0.001894) SHOW TABLES
SQL (0.004846) SHOW FIELDS FROM stories
SQL (0.004434) SHOW KEYS FROM stories
Story Columns (0.006541) SHOW FIELDS FROM stories
SQL (0.015294) BEGIN
SQL (0.006220) INSERT INTO stories (`name`, `link`) VALUES('My shiny weblog', 'http://poocs.net/')
SQL (0.000747) COMMIT
SQL (0.001369) BEGIN
SQL (0.001114) INSERT INTO stories (`name`, `link`) VALUES('SitePoint Forums', 'http://www.sitepoint.com/forums/')
SQL (0.001174) COMMIT
SQL (0.037809) SELECT count(*) AS count_all FROM stories
Story Load (0.533204) SELECT * FROM stories WHERE (stories.id = 2) LIMIT 1
Story Load (0.013437) SELECT * FROM stories
Story Load (0.012962) SELECT * FROM stories
Story Load (0.002186) SELECT * FROM stories ORDER BY id DESC LIMIT 1
Story Load (0.012872) SELECT * FROM stories WHERE (stories.`name` = 'My shiny weblog') LIMIT 1
SQL (0.001156) BEGIN
Story Destroy (0.528325) DELETE FROM stories
WHERE id = 1
SQL (0.001247) COMMIT
Story Load (0.013047) SELECT * FROM stories WHERE (stories.`name` = 'My shiny weblog') LIMIT 1
Story Load (0.012669) SELECT * FROM stories WHERE (stories.id = 1) LIMIT 1
Core:~/shovell/log scoop$

Summary

- Generating a Model
- Building a Migration
- Using the Rails console
- Note: Ignore Scaffolding